What is claimed is:

Sub A17

1. A method of inserting table-format data where field values are inserted at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said method of inserting table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said subscript,

identifying an insertion position that indicates the position of the field value to be inserted,

in said subscript conversion array, regarding said insertion position, giving an offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said array,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving an offset value that shifts the corresponding range of subscripts upward and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said array,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

2. A method of deleting table-format data where field values are deleted at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said method of deleting table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said accepted subscript,

identifying a deletion position that indicates the position of the field value to be

SubA17

deleted, and

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving an offset value that shifts the corresponding range of subscripts downward and also increments the accepted subscript,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

3. A method of updating table-format data where field values are updated at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said method of updating table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said accepted subscript,

(1) assuming the position of the field value to be updated to be the deletion position,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving an offset value that shifts the corresponding range of subscripts downward and also increments the accepted subscript, and

(2) assuming the position of the field value to be updated to be the insertion position,

in said subscript conversion array, regarding said insertion position, giving an offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said array,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving an offset value that shifts the corresponding range of subscripts upward and also decrements the accepted subscript, and

placing said field value to be updated at the stipulated position after the end of said array, and

SUB A1

said (1) and (2) or (2) and (1) are executed sequentially,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

4. A value conversion method for table-format data that converts the values of field values in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said value conversion method for table-format data is characterized in comprising the steps of:

generating a value conversion array constituted such that it gives an offset value corresponding to the range of said field value, and

being constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said field value to the field value corresponding to said subscript within said array.

5. A data structure for table-format data constituted such that the field values corresponding to an arbitrary record number are identified in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said data structure for table-format data is characterized in that:

said table-format data is constituted in a manner such that it is divided into one or more information blocks consisting of: a value list containing a first actual array in which the field values are stored in the order of a field value number corresponding to the field value belonging to a specified field, and a pointer array containing a second actual array in which pointer values for pointing to said field value numbers are stored in a unique record number order,

a second subscript conversion array constituted such that an input, output of the pointer array is given as the subscript in said value list, and a third offset corresponding to the range of said subscript is given is formed and field value within the first actual array is identified by means of the output of the pointer array given by an offset via said second subscription conversion array.

6. The data structure according to claim 5 characterized in that each of said subscript conversion arrays comprises: a start position array consisting of start positions that indicate the minimum value of the subscripts contained within each of

SUB A17

the stipulated ranges, and/or an end position array consisting of end positions that indicate the maximum value of the subscripts contained in said stipulated range, and an offset array consisting of the corresponding offset values.

7. The data structure according to claim 5 characterized in that said value conversion array comprises: a start position array consisting of start positions that indicate the minimum value of the subscripts contained within each of the stipulated ranges, and/or an end position array consisting of end positions that indicate the maximum value of the values contained in said stipulated range, and an offset array consisting of the corresponding offset values.

8. A method of inserting table-format data where field values are inserted at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said method of inserting table-format data is characterized in comprising the steps of:

(1) regarding said value list:

identifying an insertion position that indicates the position of the field value to be inserted,

in said second subscript conversion array, regarding said insertion position, giving a third offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said first array,

in said second subscript conversion array, regarding those records having a value greater than the subscript corresponding to said insertion position, giving a third offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said first actual array,

(2) regarding said pointer array,

identifying an insertion position that indicates the position of the pointer value corresponding to the record number to be inserted,

in said first subscript conversion array, regarding said insertion position, giving a first offset value that defines the corresponding subscript and also identifies a stipulated position after the end of said first actual array,

SUB A17

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving a first offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing a new pointer value greater than the existing pointer values at the stipulated position after the end of said second actual array, and

(3) regarding said pointer array,

in said value conversion array, giving a second offset value that increments those records that have a value greater than the pointer value corresponding to the insertion position in said value list,

in said value conversion array, giving a second offset value such that said new pointer value identifies a position corresponding to said insertion position.

9. The method of inserting table-format data according to claim 8 characterized in that:

the table comprises m record numbers and n field values,

when the insertion position for field values within said value list is i (0<=i<=n-1), a field value is placed at the end n of the first actual array, and the record number insertion position is j (0<=j<=m-1), a pointer value is placed at the end m of the second actual array,

(1) in the second subscript conversion array of said value list:

0 is given as the third offset value in the case that the value is in the range of (i-1) or less,

(n-1) is given as the third offset value in the case that the value is i, and

-1 is given as the third offset value in the case that the value is in the range of (i+1) or greater and n or less,

(2) in the first subscript conversion array of said pointer array:

0 is given as the first offset value in the case that the subscript is in the range of (j-1) or less,

(m-j) is given as the first offset value in the case that the subscript is j, and

(-1) is given as the first offset value in the case that the subscript is in the range of (j+1) or greater and m or less,

(3) in the value conversion array of said pointer array:

0 is given as the second offset value in the case that the value is in the range of (i-1),

1 is given as the second offset value in the case that the value is in the range of i or greater and (n-1) or less, and

(i-n) is given as the second offset value in the case that the value is n.

10. The method of inserting table-format data according to claim 8 characterized in that:

the table comprises m record numbers and n field values,

when the insertion position for field values within said value list is i (0<=i<=n-1), a field value is placed at a stipulated position z (z>=n) after the end of the first actual array, and the record number insertion position is j (0<=j<=m-1), a pointer value is placed at a stipulated position x (x>=m) after the end of the second actual array,

(1) in the second subscript conversion array of said value list:

0 is given as the third offset value in the case that the value is in the range of (i-1) or less,

(z-1) is given as the third offset value in the case that the value is i, and

(-1) is given as the third offset value in the case that the value is in the range of (i+1) or greater and n or less,

(2) in the first subscript conversion array of said pointer array:

0 is given as the first offset value in the case that the subscript is in the range of (j-1) or less,

(x-j) is given as the first offset value in the case that the subscript is j, and

(-1) is given as the first offset value in the case that the subscript is in the range of (j+1) or greater and m or less, and also

(3) in the value conversion array of said pointer array:

0 is given as the second offset value in the case that the value is in the range of (i-1),

1 is given as the second offset value in the case that the value is in the range of i or greater and (n-1) or less, and

(i-y) is given as the second offset value in the case that the value is y (where y is the pointer value stored at the position x of the second actual array).

11. A method of deleting table-format data where field values are deleted at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said method of deleting table-format data is characterized in comprising the steps of:

regarding said pointer array,

identifying a deletion position that indicates the position of the pointer value to be deleted,

in said first subscript conversion array, regarding said insertion position, defining the corresponding subscript,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving a first offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript.

12. The method of deleting table-format data according to claim 11 characterized in that:

the table comprises m record numbers and n field values,

when the record number deletion position is j (0<=j<=m-1),

in the first subscript conversion array of said pointer array:

0 is given as the first offset value in the case that the subscript is in the range of (j-1) or less, and

1 is given as the offset value in the case that the subscript is in the range of j or greater and (m-2) or less.

Sub A7

13. A method of updating table-format data where field values are updated at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said method of updating table-format data is characterized in comprising the steps of:

(A) (1) regarding said value list:

assuming the position of the field value to be updated to be the insertion position, identifying an insertion position that indicates the position of the field value to be inserted,

in said second subscript conversion array, regarding said insertion position, giving a third offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said first array,

in said second subscript conversion array, regarding those records having a value greater than the subscript corresponding to said insertion position, giving a third offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said first actual array,

(2) regarding said pointer array,

assuming the pointer value to be updated to be the pointer value to be inserted, identifying an insertion position that indicates the position of the pointer value to be inserted,

in said first subscript conversion array, regarding said insertion position, giving a first offset value that defines the corresponding subscript and also identifies a stipulated position after the end of said first actual array,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving a first offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript,

placing a new pointer value greater than the existing pointer values at the stipulated position after the end of said second actual array, and

(3) regarding said pointer array,

SUB A17

in said value conversion array, giving a second offset value that increments those records that have a value greater than the pointer value corresponding to the insertion position in said value list,

in said value conversion array, giving a second offset value such that said new pointer value identifies a position corresponding to said insertion position,

(B) regarding said pointer array,

identifying the position of the pointer value to be updated as the deletion position, in consideration of said insertion position,

in said first subscript conversion array, regarding said deletion position, defining the corresponding subscript,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving a first offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript, and

either of said (A) and (B) or said (B) and (A) are executed sequentially.

14. The method of updating table-format data according to claim 13 characterized in that:

the table comprises m record numbers and n field values,

when the updating position for field values within said value list is i (0<=i<=n-1), a field value is placed at the end n of the first actual array, and the record number updating position is j (0<=j<=m-1), a pointer value is placed at the end m of the second actual array,

(1) in the second subscript conversion array of said value list:

0 is given as the third offset value in the case that the value is in the range of (i-1) or less,

(n-1) is given as the third offset value in the case that the value is i, and

(-1) is given as the third offset value in the case that the value is in the range of (i+1) or greater and n or less,

(2) in the first subscript conversion array of said pointer array:

SUB A7

0 is given as the first offset value in the case that the subscript is in the range of (j-1) or less,

(m-j) is given as the first offset value in the case that the subscript is j, and

0 is given as the first offset value in the case that the subscript is in the range of (j+1) or greater and (m+1) or less,

(3) in the value conversion array of said pointer array:

0 is given as the second offset value in the case that the value is in the range of (i-1),

1 is given as the second offset value in the case that the value is in the range of i or greater and (n-1) or less, and

(i-n) is given as the second offset value in the case that the value is n.

15. The method of updating table-format data according to claim 13 characterized in that:

the table comprises m record numbers and n field values,

when the insertion position for field values within said value list is i (0<=i<=n-1), a field value is placed at a stipulated position z (z>=n) after the end of the first actual array, and the record number insertion position is j (0<=j<=m-1), a pointer value is placed at a stipulated position x (x>=m) after the end of the second actual array,

(1) in the second subscript conversion array of said value list:

0 is given as the third offset value in the case that the value is in the range of (i-1) or less,

(z-1) is given as the third offset value in the case that the value is i, and

(-1) is given as the third offset value in the case that the value is in the range of (i+1) or greater and n or less,

(2) in the first subscript conversion array of said pointer array:

0 is given as the first offset value in the case that the subscript is in the range of (j-1) or less,

(x-j) is given as the first offset value in the case that the subscript is j, and

0 is given as the first offset value in the case that the subscript is in the range of (j+1) or greater and (m+1) or less, and also

(3) in the value conversion array of said pointer array:

0 is given as the second offset value in the case that the value is in the range of (i-1),

1 is given as the second offset value in the case that the value is in the range of i or greater and (n-1) or less, and

(i-y) is given as the second offset value in the case that the value is y (where y is the pointer value stored at the position x of the second actual array).

16. A method of deleting table-format data where field values are deleted at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said method of deleting table-format data is characterized in comprising the steps of:

regarding said record number, providing a fourth actual array in which the record numbers themselves are placed, and a third subscript conversion array given stipulated offset values according to the range of subscripts for identifying said record number,

identifying a deletion position that indicates the position of said record number to be deleted,

in said third subscript conversion array, defining a corresponding subscript regarding said deletion,

in said third subscript conversion array, regarding those records having a value greater than the subscript corresponding to said deletion position, giving a fourth offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript.

17. A method of deleting table-format data where field values are deleted at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said method of deleting table-format data is characterized in comprising the steps of:

SUBA7

regarding said record number, providing an array in which the record numbers themselves are placed, and a subscript conversion array given stipulated offset values according to the range of subscripts for identifying said record number,

identifying a deletion position that indicates the position of said record number to be deleted,

in said subscript conversion array, defining a corresponding subscript regarding said deletion, and

in said subscript conversion array, regarding those records having a value greater than the subscript corresponding to said deletion position, giving an offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript.

18. A transaction processing method characterized in that, when the insertion, deletion and/or updating of data in table-format data is performed by means of a method according to any of claims 8-15, and either a rollback or commit is performed, said subscript conversion array and value conversion array are discarded.

19. A transaction processing method characterized in that, when the insertion, deletion and/or updating of data in table-format data is performed by means of a method according to any of claims 8-15, and a rollback is performed, said subscript conversion array and value conversion array are discarded.

20. A parallel processing method where a plurality of processes are performed simultaneously on table-format data having the data structure recited in any of claims 5-7, wherein

among the aforementioned processes, those that involve the insertion, deletion and/or updating of data are constituted such that, in said pointer array, they go through a first subscript conversion array, second actual array and value conversion array, and also, in said value list, they go through a second subscript conversion array and a first actual array, and

on the other hand, among the aforementioned processes, those that do not involve the insertion, deletion and/or updating of data are constituted such that, in said pointer array, they go through a second subscript conversion array, and also, in said value list, they go through a first actual array.

21. The parallel processing method according to claim 20 characterized in that said processes that involve the insertion, deletion and/or updating of data utilize a method according to any of claims 8-15.

Sub A1

22. A record locking method for table-format data having the data structure recited in any of claims 5-7, wherein

an information block containing an array for controlling locking is provided, and

field values corresponding to each of said record numbers that indicate the type of locking are placed within the array of said information block.

23. A computer-readable recording medium recorded with a data insertion program for table-format data where field values are inserted at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said data insertion program for table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said subscript,

identifying an insertion position that indicates the position of the field value to be inserted,

in said subscript conversion array, regarding said insertion position, giving an offset value that defines the range of the corresponding subscript and also identifies the end of said array,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving an offset value that shifts the corresponding range of subscripts upward and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said array,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

24. A computer-readable recording medium recorded with a data deletion program for table-format data where field values are deleted at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

Sub A7

said data deletion program for table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said accepted subscript,

identifying a deletion position that indicates the position of the field value to be deleted, and

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving an offset value that shifts the corresponding range of subscripts downward and also increments the accepted subscript,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

25. A computer-readable recording medium recorded with a data updating program for table-format data where field values are updated at arbitrary positions in table-format data represented by an array of records containing a field and the field values contained therein, wherein

said data updating program for table-format data is characterized in comprising the steps of:

generating a subscript conversion array constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said accepted subscript,

(1) assuming the position of the field value to be updated to be the deletion position,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving an offset value that shifts the corresponding range of subscripts downward and also increments the accepted subscript, and

(2) assuming the position of the field value to be updated to be the insertion position,

in said subscript conversion array, regarding said insertion position, giving an offset value that defines the range of the corresponding subscript and also identifies

Sub A1

the end of said array,

in said subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving an offset value that shifts the corresponding range of subscripts upward and also decrements the accepted subscript, and

placing said field value to be updated at the stipulated position after the end of said array, and

said (1) and (2) or (2) and (1) are executed sequentially,

such that an offset value according to the range of subscripts within the subscript conversion array is given as said subscript and the field value with the array is identified by means of the subscript given by said offset value.

26. A computer-readable recording medium recorded with a value conversion program for table-format data that converts the values of field values of table-format data represented by an array of records containing a field and the field values contained therein, wherein

said value conversion program for table-format data is characterized in comprising the steps of:

generating a value conversion array constituted such that it gives an offset value corresponding to the range of said field value, and

being constituted such that it accepts a record number as a subscript and gives an offset value corresponding to the range of said field value to the field value corresponding to said subscript within said array.

27. A computer-readable recording medium recorded with a data insertion program for table-format data where field values are inserted at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said data insertion program for table-format data is characterized in comprising the steps of:

(1) regarding said value list:

identifying an insertion position that indicates the position of the field value to be inserted,

in said second subscript conversion array, regarding said insertion position,

SubA17

giving a third offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said first array,

in said second subscript conversion array, regarding those records having a value greater than the subscript corresponding to said insertion position, giving a third offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said first actual array,

(2) regarding said pointer array,

identifying an insertion position that indicates the position of the pointer value corresponding to the record number to be inserted,

in said first subscript conversion array, regarding said insertion position, giving a first offset value that defines the corresponding subscript and also identifies a stipulated position after the end of said first actual array,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving a first offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing a new pointer value greater than the existing pointer values at the stipulated position after the end of said second actual array, and

(3) regarding said pointer array,

in said value conversion array, giving a second offset value that increments those records that have a value greater than the pointer value corresponding to the insertion position in said value list,

in said value conversion array, giving a second offset value such that said new pointer value identifies a position corresponding to said insertion position.

28. A computer-readable recording medium recorded with a data deletion program for table-format data where field values are deleted at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said data deletion program for table-format data is characterized in comprising the steps of:

SUBA17

regarding said pointer array,

identifying a deletion position that indicates the position of the pointer value to be deleted,

in said first subscript conversion array, regarding said insertion position, defining the corresponding subscript,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving a first offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript.

29. A computer-readable recording medium recorded with a data updating program for table-format data where field values are updated at arbitrary positions in table-format data represented by the data structure recited in any of claims 5-7, wherein

said data updating program for table-format data is characterized in comprising the steps of:

(A) (1) regarding said value list:

assuming the position of the field value to be updated to be the insertion position, identifying an insertion position that indicates the position of the field value to be inserted,

in said second subscript conversion array, regarding said insertion position, giving a third offset value that defines the range of the corresponding subscript and also identifies a stipulated position after the end of said first array,

in said second subscript conversion array, regarding those records having a value greater than the subscript corresponding to said insertion position, giving a third offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript, and

placing said field value to be inserted at the stipulated position after the end of said first actual array,

(2) regarding said pointer array,

assuming the pointer value to be updated to be the pointer value to be inserted, identifying an insertion position that indicates the position of the pointer value to be inserted,

SUB A17

in said first subscript conversion array, regarding said insertion position, giving a first offset value that defines the corresponding subscript and also identifies a stipulated position after the end of said first actual array,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said insertion position, giving a first offset value that shifts the corresponding range of subscripts upward so that the values that define said range become larger and also decrements the accepted subscript,

placing a new pointer value greater than the existing pointer values at the stipulated position after the end of said second actual array, and

(3) regarding said pointer array,

in said value conversion array, giving a second offset value that increments those records that have a value greater than the pointer value corresponding to the insertion position in said value list,

in said value conversion array, giving a second offset value such that said new pointer value identifies a position corresponding to said insertion position,

(B) regarding said pointer array,

identifying the position of the pointer value to be updated as the deletion position, in consideration of said insertion position,

in said first subscript conversion array, regarding said deletion position, defining the corresponding subscript,

in said first subscript conversion array, regarding those records having a record number greater than the record number corresponding to said deletion position, giving a first offset value that shifts the corresponding range of subscripts downward so that the values that define said range become smaller and also increments the accepted subscript, and

either of said (A) and (B) or said (B) and (A) are executed sequentially.